Refining Software Quality Prediction with LOD

Davide Ceolin[1], Till Döhmen[1], and Joost Visser[2]

[1] VU University Amsterdam
de Boelelaan 1081
1081HV Amsterdam, The Netherlands
d.ceolin@vu.nl
[2] Software Improvement Group
Rembrandt Toren, 15th floor, Amstelplein 1
1096 HA Amsterdam, The Netherlands

Abstract. The complexity of software systems is growing and the computation of several software quality metrics is challenging. Therefore, being able to use the already estimated quality metrics to predict their evolution is a crucial task. In this paper, we outline our idea to use Linked Open Data to enrich the information available for such prediction. We report our experience so far, and we outline the preliminary results obtained.

1 Introduction

Software size and complexity is growing, thus being able to estimate and predict software quality is crucial to monitor the process of software development and promptly steer it. In fact, a quality metric provides a value summarizing one relevant aspect of the software that can be consulted to identify issues or risks in the development process or in the software itself. Therefore, several different quality dimensions have been defined, as described, for instance, by Kan [8].

Estimating software quality is then a crucial but challenging task, for several reasons including the complexity of the software to be measured and the fact that these measures are often hard to quantify: some of them depend on runtime software behavior, some on static software properties. The estimation of the values of these measures is possible, as demonstrated, for instance, by Alves and Visser [2] and Bouwers [4]. However, given the complexity of this task, we propose to use such estimates to predict the temporal evolution of these values.

Preliminary analyses on a dataset from the Software Improvement Group[3] show encouraging results on the use of these estimates as starting point for the prediction of the evolution over time of software quality ratings.[4] We hypothesize that, by using Linked Open Data (LOD) we can improve and refine the accuracy of our predictions. In particular, by enriching the information available about the projects analyzed, we can categorize these projects (e.g., by industry sector or programming language), thus increasing the possibility to group

[3] http://www.sig.eu
[4] For confidentiality reasons, we could not make the dataset publicly available.

together projects showing similar quality evolution over time. We present here some preliminary encouraging results obtained in this direction, and we discuss a series of open issues that we need to address in order to extend this research.

The rest of this paper is structured as follows: Section 2 introduces related work. Section 3 describes the enrichment of software projects data. Section 4 provides preliminary results, that are discussed in Section 5.

2 Related Work

Software quality prediction is an important issue, that has been tackled from different points of view. As Al-Jamini and Ahmed [1] describe in their review, several relevant approaches to this problem make use of machine learning.

We have also employed machine learning techniques (in particular, Markov chains [12]) to predict software quality based on the starting rating of a project [5]. The results are promising and we will aim at perfecting them with additional features, properly selected from external sources, like LOD. The future quality value of systems shows a strong correlation with the current quality rating, due to the fact that the rating usually changes very slowly over time. Moreover, a second trend was discovered which revealed that higher quality systems tend to deteriorate in quality and low-quality systems tend to improve, both with the tendency towards the medium quality level. This could be explained as a case of regression towards the mean [6], i.e., could be due to noise in the extreme quality ratings that disappears as more accurate estimates are provided. However, this possible explanation still needs to be evaluated and, anyway, could explain only the second trend. These two trends, for very high or very low-quality systems, yield a high uncertainty in the prediction. Using LOD, we expect to obtain more tailored predictions (e.g., by identifying software quality trends associated to the programming language adopted) to reduce prediction uncertainty.

Misirli et al.[11] propose the use of Bayesian Networks to make software quality predictions. As the number of potentially useful features grows (consequently to LOD enrichment), we will consider this approach in the future. Jing et al. [7] use a dictionary learning-approach that represents a more specialized but limited approach as compared to our use of LOD.

3 Enriching Software Quality Prediction with LOD

Our hypothesis is that by enriching the information about the projects we analyze with LOD, we can obtain features that are useful for improving the software quality prediction. For instance, software quality could vary in different industrial sectors or the programming language used could affect quality evolution.

Our focus is on a dataset provided by the Software Improvement Group, which consists mainly of projects of Dutch companies and of a few additional European customers. We enriched the dataset using mainly DBpedia [3]. In the enrichment process, we encountered the following issues:

Missing information DBpedia contains a description of only 209 companies located in the Netherlands. Additional companies have been identified in the Dutch DBpedia[5], which contains the description of 3.883 companies, but does not provide information about their location.

Disambiguation Some companies have homonyms. To disambiguate resources and identify the right URI for a given company, we expect to employ heuristics based on the company website, its location, and industry sector.

Consistency literals vs. URIs Some classifications are available in an inconsistent manner. For instance, industry can appear both as `http://dbpedia.org/ontology/industry` and `http://dbpedia.org/property/industry`. In some cases, the value of one of these two properties is reported only as a literal value, thus affecting the possibility to perform ontological reasoning.

4 Preliminary Results

We performed a preliminary analysis on a dataset consisting of 1019 snapshots of maintainability of 112 companies. These snapshots already presented a first industry classification provided by SIG. In total, 14 industrial sectors are present.

We computed the semantic similarity between each possible combination of industrial categories using the Wikipedia distance [10] and the WU & Palmer distance [17]. On these data, we performed a series of preliminary analyses:

1. We run a Wilcoxon signed-rank test [16] at 95% confidence level to check if the observations are significantly different when grouped per industrial sector. These results show a weak positive Spearman [15] correlation with both the Wikipedia (0.07) and the Wu & Palmer (0.14) distances.
2. We computed the same procedure as above by using also the Kolmogorov-Smirnov test [9, 14] . This resulted in a slightly higher correlation, 0.16 for the Wikipedia distance and 0.24 for the Wu & Palmer distance.
3. We computed the contrast analysis [13] of the linear combinations of the observations, again grouped per industrial sector. The resulting contrast estimators showed a weak correlation with the Wikipedia distance (0.15) and with the Wu & Palmer distance values (0.12).
4. We grouped a small set of observations aligned with DBpedia by industrial sector of the companies involved (telecommunication and financial services). According to a Wilcoxon signed-rank test at 90% significance, the two groups are significantly different, according to the Kolmogorov-Smirnov test, not.

5 Discussion and Future Work

We present an early stage work about the use of LOD to refine the precision and accuracy of software quality prediction. We performed a series of exploratory and preliminary studies which shows a low correlation between the maintainability

[5] `http://nl.dbpedia.org`

and the industry sector of these projects. These results provide the basis for further exploration because: (1) the existence of a weak correlation is confirmed by more tests, hence it is possible that we can identify a subset of the data analyzed that presents a higher correlation; (2) the different methods for computing semantic similarity and different statistical significance tests provided significantly different results, thus indicating the need for exploring different computational techniques; (3) as shown by the last item of Section 4, the industrial sector seems to be a discriminant for software quality, although this aspect needs to be evaluated on larger datasets; and (5) our analyses focused on a limited set of enrichment features, but several others are utilizable. So, we plan to extend this research to identify the most robust methods to perform these predictions, and we will extend these analyses including additional LOD features and sources.

Acknowledgements This work is funded by Amsterdam Data Science.

References

1. H. Al-Jamimi and M. Ahmed. Machine learning-based software quality prediction models: State of the art. In *ICISA*, pages 1–4, 2013.
2. T. L. Alves and J. Visser. Static estimation of test coverage. In *SCAM*, pages 55–64. IEEE Computer Society, 2009.
3. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. In *ISWC*, volume 4825, pages 722–735. Springer, 2007.
4. E. Bouwers, J. P. Correia, A. van Deursen, and J. Visser. Quantifying the analyzability of software architectures. In *WICSA*, pages 83–92. IEEE, 2011.
5. T. Döhmen, D. Ceolin, and J. Visser. Towards Building a Software Quality Prediction Model. Technical report, Software Improvement Group, 2015.
6. F. Galton. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246–263, 1886.
7. X.-Y. Jing, S. Ying, Z.-W. Zhang, S.-S. Wu, and J. Liu. Dictionary learning based software defect prediction. In *ICSE*, pages 414–423, 2014.
8. S. Kan. *Metrics and Models in Software Quality Engineering*. Pearson, 2002.
9. A. Kolmogorov. Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, 4:1–11, 1933.
10. D. Milne and I. H. Witten. An open-source toolkit for mining wikipedia. *Artif. Intell.*, 194:222–239, 2013.
11. A. T. Misirli and A. B. Bener. A mapping study on bayesian networks for software quality prediction. In *RAISE*, pages 7–11, 2014.
12. J. R. Norris. *Markov chains*. Cambridge University Press, 1998.
13. R. Rosenthal and R. L. Rosnow. *Contrast analysis : focused comparisons in the analysis of variance*. Cambridge University press, 1985.
14. N. Smirnov. Table for Estimating the Goodness of Fit of Empirical Distributions. *The Annals of Mathematical Statistics*, 19(2):279–281, 1948.
15. C. Spearman. The proof and measurement of association between two things. *Amer. J. Psychol.*, 15:72101, 1904.
16. F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1:80–83, 1945.
17. Wu, Z. and Palmer, M. Verb semantics and lexical selection. In *ACL*. ACL, 1994.